



### WHAT WE'RE DOING:

Up to this point we have focused entirely on outputs, time to get our Arduino to listen, watch and feel. We'll start with a simple pushbutton. Wiring up the pushbutton is simple. There is one component, the pull up resistor, that might seem out of place.

This is included because an Arduino doesn't sense the same way we do (ie button pressed, button unpressed). Instead it looks at the voltage on the pin and decides whether it is HIGH or LOW. The button is set up to pull the Arduino's pin LOW when it is pressed, however, when the button is unpressed the voltage of the pin will float (causing occasional errors). To get the Arduino to reliably read the pin as HIGH when the button is unpressed, we add the pull up resistor.

(note: the first example program uses only one of the two buttons)

### THE CIRCUIT:

#### Parts:



**CIRC-07  
Breadboard Sheet  
x1**



**2 Pin Header  
x4**



**Pushbutton  
x2**



**Wire**



**10k Ohm Resistor  
Brown-Black-Orange  
x2**

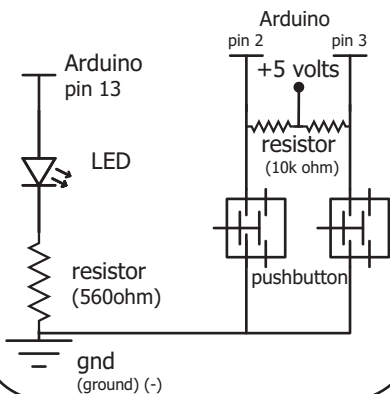


**560 Ohm Resistor  
Green-Blue-Brown  
x1**



**Red LED  
x1**

#### Schematic



#### The Internet

..download..

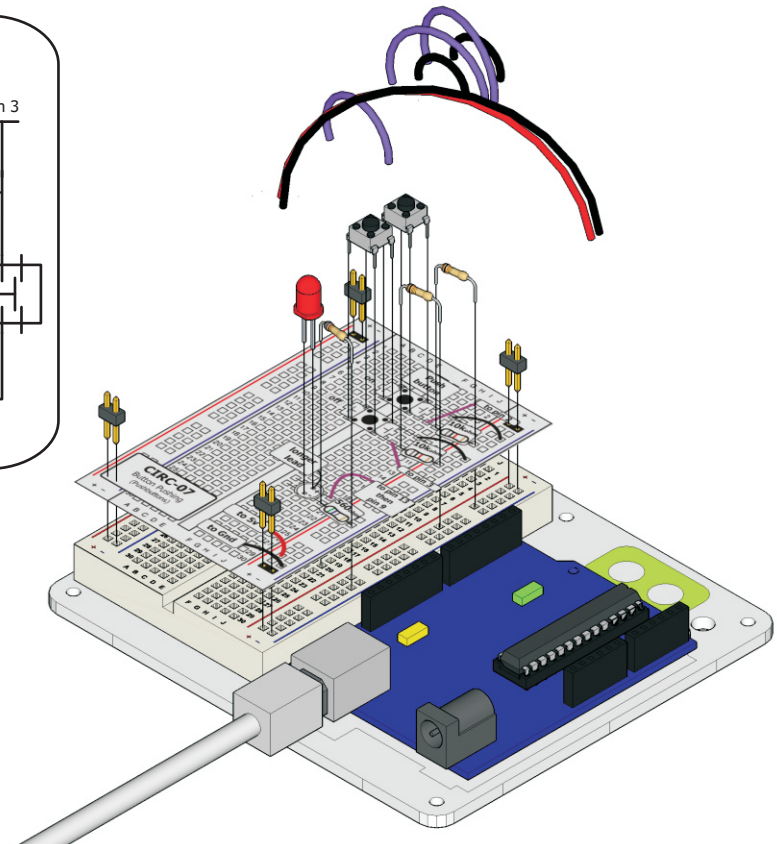
breadboard layout sheet

<http://ardx.org/BBL507>

..view..

assembly video

<http://ardx.org/VIDE07>



**CODE** (no need to type everything in just click)**File > Examples > 2.Digital > Button**

(example from the great arduino.cc site, check it out for other great ideas)

```

/*
 * Button
 * by DojoDave <http://www.0j0.org>
 *
 * Turns on and off a light emitting diode(LED) connected to digital
 * pin 13, when pressing a pushbutton attached to pin 7.
 * http://www.arduino.cc/en/Tutorial/Button
 */
int ledPin = 13;           // choose the pin for the LED
int inputPin = 2;          // choose the input pin (for a pushbutton)
int val = 0;               // variable for reading the pin status

void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare pushbutton as input
}

void loop(){
  val = digitalRead(inputPin); // read input value
  if (val == HIGH) {           // check if the input is HIGH
    digitalWrite(ledPin, LOW); // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}

```

**NOT WORKING?** (3 things to try)**Light Not Turning On**

The pushbutton is square and because of this it is easy to put it in the wrong way. Give it a 90 degree twist and see if it starts working.

**Light Not Fading**

A bit of a silly mistake we constantly made, when you switch from simple on off to fading remember to move the LED wire from pin 13 to pin 9.

**Underwhelmed?**

No worries these circuits are all super stripped down to make playing with the components easy, but once you throw them together the sky is the limit.

**MAKING IT BETTER****On button off button:**

The initial example may be a little underwhelming (ie. I don't really need an Arduino to do this), let's make it a little more complicated. One button will turn the LED on the other will turn the LED off. Change the code to:

```

int ledPin = 13;           // choose the pin for the LED
int inputPin1 = 3;         // button 1
int inputPin2 = 2;         // button 2

void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin1, INPUT); // make button 1 an input
  pinMode(inputPin2, INPUT); // make button 2 an input
}

void loop(){
  if (digitalRead(inputPin1) == LOW) {
    digitalWrite(ledPin, LOW); // turn LED OFF
  } else if (digitalRead(inputPin2) == LOW) {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}

```

Upload the program to your board, and start toggling the LED on and off.

**Fading up and down:**

Lets use the buttons to control an analog signal. To do this you will need to change the wire connecting the LED from pin 13 to pin 9, also change this in code.

```

int ledPin = 13; ----> int ledPin = 9;
Next change the loop() code to read.

int value = 0;
void loop(){
  if (digitalRead(inputPin1) == LOW) { value--; }
  else if (digitalRead(inputPin2) == LOW) { value++; }
  value = constrain(value, 0, 255);
  analogWrite(ledPin, value);
  delay(10);
}

```

**Changing fade speed:**

If you would like the LED to fade faster or slower, there is only one line of code that needs changing;

```

delay(10); ----> delay(new #);
To fade faster make the number smaller, slower requires a
larger number.

```

**MORE, MORE, MORE:**

More details, where to buy more parts, where to ask more questions:

<http://ardx.org/CIRC07>